NOOB HowTo

A piece of history

The Naive Objectionable Opaque Bank (NOOB) was founded by the dutchman Piet Poenschuiver in 2018. Originally named Naïeve Ontzettend Onveilige Bank, the NOOB played a crucial role in stealing money from the poor and giving it to the rich, much like Robin Hood. Piet became one of the first clients of the NOOB, which, by intimate knowledge of its inner workings, made him a very wealthy man.

Mission statement

The NOOB is a pure clearing house:

- It doesn't store anything (except for blackmailing purposes)
- It doesn't charge anything (that's left to its trusted affiliates)
- It doesn't check anything (especially not how the money flows)

How it works

To get acquainted with how it works:

- Start the NOOB by running noob.py
- Start several instances of the consumer bank emulator (NOOC), each with its own bank code and currency conversion rate [euro's per coin] as command line parameters, e.g. by running nooc.py ingb 1
- Note that each client must have a unique bank code (you may use e.g. INGB_NL and INGB_GB if a bank has multiple locations with different currencies
- Play around with both, and take a peek at the source code if you're interested

Note that while the NOOB is written in Python 3.7, its clients may be written in any language, as long as they speak JSON over WebSockets.

Clients are required to utilize the NOOB protocol and play two roles at the same time, 'master' and 'slave'.

- A master accepts user commands and executes them locally or on a remote bank.
- A slave accepts commands from a remote bank and executes them locally.

Each role is represented by a separate websocket connection via the same IP address, including portnumber. The NOOB is always the server and a NOOC is always a client to that server. You can run the NOOB and multiple NOOC's on one machine for development purposes, by using 'localhost'. To perform a transaction, both local and remote NOOC have to be online in addition to the NOOB. **You can perform remote transactions via the NOOB on several NOOC clients to test your own consumer bank, that may be written e.g. in C++ or Java, the latter being common in the banking world.**

Suppose for example:

- The target bank code is INGB (case insensitive)
- The target account number is 300400
- The target pin code is 1234

Sending the command *["register", "master", "INGB"]* to the NOOB as first command over a newly made socket connection will register that connection with the NOOB for the INGB master role.

Sending the command *["register", "slave", "INGB"]* to the NOOB as first command over a newly made socket connection will register that connection with the NOOB for the INGB slave role.

Master role

Remotely opening an ABNA account from bank INGB happens by sending to the NOOB:

["ABNA", "open", 300400, 1234, 0]

Remotely closing this ABNA account from bank INGB happens by sending to the NOOB:

["ABNA", "close", 300400, 1234, 0]

Deposing 1000 local coins on this ABNA account from bank INGB happens by sending to the NOOB:

["ABNA", "deposit", 300400, 1234, 1000]

Withdrawing 199.50 local coins from this ABNA account from bank INGB happens by sending to the NOOB:

["ABNA", "withdraw", 300400, 1234, 199.50]

The *hack* command allows inspecting all local accounts on a NOOC.

Slave role

All commands received by the slave are the same as the ones sent by the master, only with the first (bank code) parameter omitted.

Replies

All commands for both roles are replied upon:

[True] means succes, *[False]* means failure.

Our motto: your failure is our success!

Yours greedily

Piet Poenschuiver